

Môn thi: Cấu trúc dữ liệu và giải thuật
Mã lớp: Các lớp IT003 - Hệ đại trà, chất lượng cao
Thời gian làm bài: 90 phút
(*Sinh viên không được sử dụng tài liệu*)

Câu 1: (2.5 điểm)

- a. Trình bày các bước giải thuật sắp xếp Quick sort (không viết chương trình) để sắp xếp mảng số nguyên N phần tử **giảm dần**, cho biết độ phức tạp giải thuật. (1.5 điểm)

Thang điểm đề nghị :

Ý 1 : Trình bày giải thuật : 1 điểm

- SV trình bày đúng các bước giải thuật: 1 điểm.
- SV viết hàm cài đặt (đúng) thay cho các bước giải thuật : trừ 0.5 điểm

Ý 2 : Trình bày độ phức tạp giải thuật : 0.5 điểm

- SV trình bày chính xác độ phức tạp giải thuật : 0.5 điểm

- b. Trình bày các bước (vẽ từng bước) áp dụng giải thuật trong câu 1.a để sắp xếp mảng số nguyên {10, 5, 30, 70, 40, 80, 90} giảm dần. (1 điểm)

Mảng ban đầu :

10	5	30	70	40	80	90
----	---	----	----	----	----	----

Lần 1 : Chọn phần tử 70 làm phần tử phân hoạch:

90	80	70	30	40	5	10
----	----	----	----	----	---	----

Lần 2 : Chọn phần tử 80 làm phần tử phân hoạch [đoạn 90,80]

90	80	70	30	40	5	10
----	----	----	----	----	---	----

Lần 3 : Chọn phần tử 5 làm phần tử phân hoạch [đoạn 30,40,5,10]

90	80	70	30	40	10	5
----	----	----	----	----	----	---

Lần 4 : Chọn phần tử 40 làm phần tử phân hoạch [đoạn 30,40,10]

90	80	70	40	30	10	5
----	----	----	----	----	----	---

Kết thúc : Mảng sắp giảm : 90, 80, 70, 40, 30, 10, 5

Thang điểm đề nghị :

- Vẽ chính xác các bước sắp dãy số giảm dần: 1 điểm.

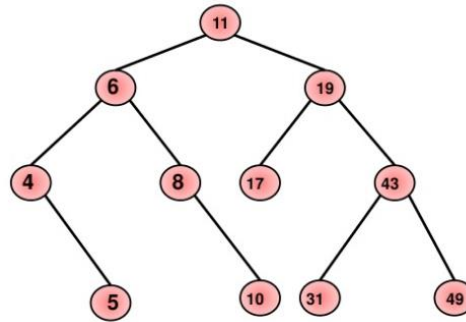
Câu 2: (4 điểm)

Cho dãy số sau: 11, 6, 8, 19, 4, 10, 5, 17, 43, 49, 31

Hãy thực hiện các yêu cầu sau:

MSSV:.....

- a. Xây dựng **cây nhị phân tìm kiếm** từ dãy số đã cho vào cây theo thứ tự thêm các số từ trái sang phải của dãy số (1 điểm)



Thang điểm đề nghị :

- Vẽ chính xác cây hoàn chỉnh : được trọn 1 điểm.
- Vẽ sai 1 số trên cây -> cây không còn là cây nhị phân tìm kiếm : không có điểm.
- Tạo cây với thứ tự sai của dãy số đã cho : không có điểm.

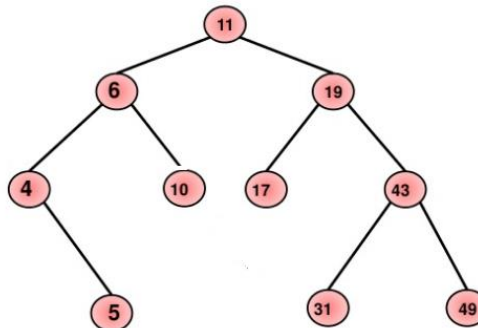
- b. Duyệt cây trong câu 2.a theo Node-Left-Right, Right-Left-Node (1 điểm)

Thang điểm đề nghị :

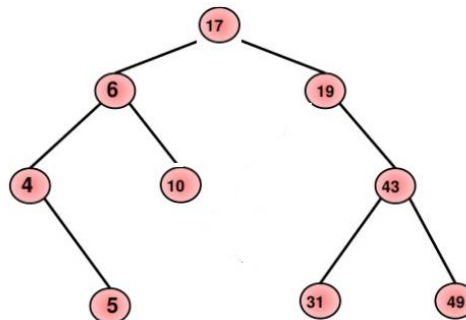
- Node – Left – Right viết đúng : 0,5 điểm – 11,6,4,5,8,10,19,17,43,31,49
- Right – Left - Node viết đúng : 0,5 điểm – 49,31,43,17,19,10,8,5,4,6,11

- c. Xóa khỏi cây **lần lượt các nút 8, 11, 43, 6** (vẽ hình từng trường hợp) sao cho cây vẫn là cây nhị phân tìm kiếm sau khi xóa nút. (1 điểm)

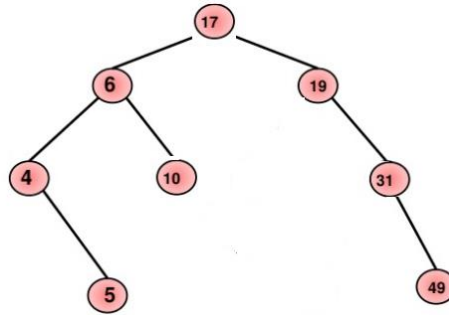
+ Xóa nút 8 :



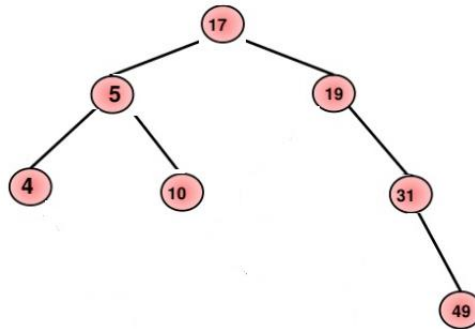
+ Xóa nút 11:



+ Xóa nút 43 :



+ Xóa nút 6 :



Thang điểm đề nghị :

- Xóa từng số đúng và vẽ lại cây chính xác sau khi xóa số : 0,25 điểm / 1 số thực hiện đúng.
- Không xóa lần lượt từng số và vẽ hình từng trường hợp, chỉ vẽ hình cây kết quả cuối cùng sau khi xóa cả 4 số chính xác : trừ 0.5 điểm.

d. Viết hàm in ra màn hình các nút trên cây có duy nhất một nút con (1 điểm)

```

void Print(Tree T)
{ if (T!=NULL)
  { if ((T->pLeft != NULL && T->pRight == NULL) || (T->pLeft == NULL
                                                    && T->pRight != NULL))

    cout << T->key;
    Print(T->pLeft);
    Print(T->pRight);
  }
}
  
```

Thang điểm đề nghị :

- Hoàn thành chính xác hàm : được trọn 1 điểm.
- Thiếu 1 điều kiện (con trỏ trái của nút khác NULL và con trỏ phải là NULL) HOẶC (con trỏ trái của nút là NULL và con trỏ phải khác NULL) : trừ 0.5 điểm

Câu 3: (2 điểm)

Cho bảng băm A kích thước 13 phần tử và tập khóa $K = \{10, 26, 52, 76, 13, 8, 3, 33, 60, 42\}$, ta cần nạp các giá trị khóa K vào bảng băm A sử dụng hàm băm $H(K) = K \% 13$.

MSSV:.....

Hãy vẽ bảng băm khi **thêm từng khóa K vào bảng A**, trong trường hợp xảy ra đụng độ, sử dụng phương pháp dò tuyến tính để giải quyết đụng độ.

Vị trí	Giá trị lưu	Đụng độ ?
12		
11	76	
10	10	
9	60	Đụng độ 60% $13 = 8$
8	8	
7	33	
6		
5		
4	42	Đụng độ 42% 13 $= 3$
3	3	
2	13	Đụng độ 13% 13 $= 0$
1	52	Đụng độ 52% 13 $= 0$
0	26	

Thang điểm đề nghị :

- 4 số bị đụng độ 52, 13, 42, 60 đưa vào bảng băm chính xác : 0.25 điểm cho 1 số.
- 6 số không bị đụng độ 26, 3, 33, 8, 10, 76 đưa vào bảng băm chính xác : 0.125 điểm cho 1 số.
- Hoàn thành chính xác bảng băm : được trọn 2 điểm.

Câu 4: (1.5 điểm)

Mạng xã hội là là dịch vụ kết nối các thành viên, người dùng trên Internet lại với nhau dựa theo những tiêu chí, sở thích nào đó, với nhiều mục đích khác nhau, là nơi trao đổi thông tin, chia sẻ suy nghĩ, ý tưởng mà không bị giới hạn về không gian và thời gian. Các thành viên giao tiếp với các thành viên khác trong mạng, mỗi thành viên sẽ là một chủ thể trao đổi thông tin và tương tác với người khác.

Bạn được giao nhiệm vụ xây dựng một mạng xã hội thu nhỏ với các chức năng : kết bạn với nhau, xem thông tin của bạn của mình post trên mạng và like một post của bạn giống như cách trên Facebook, hãy đề xuất ý tưởng :

- a. Mô tả cấu trúc dữ liệu bạn nghĩ sẽ sử dụng phù hợp với các chức năng của mạng đã mô tả ở phần trên. (0.75 điểm)
 - Mục tiêu câu hỏi là kiểm tra khả năng vận dụng kiến thức của sinh viên đã học để giải quyết một bài toán cụ thể.
 - SV chỉ cần đề xuất ý tưởng sử dụng cấu trúc dữ liệu đã học để giải quyết và đề xuất ý tưởng các bước giải thuật để thực hiện yêu cầu của câu hỏi trên cơ sở tính phù hợp, hợp lý với cấu trúc dữ liệu sử dụng.

MSSV:.....

-Câu hỏi KHÔNG yêu cầu về tính tối ưu của ý tưởng / giải pháp SV đề nghị sử dụng.

Thang điểm đề nghị :

- 0.25 điểm : SV thể hiện qua cách trả lời của mình : nhận diện được / xác định được các ý tưởng ban đầu :
 - Cần tổ chức, lưu trữ được danh sách thành viên trong mạng
 - Cần tổ chức, lưu trữ được các post của 1 thành viên trong mạng
 - Cần tổ chức, lưu trữ được danh sách các “bạn” của 1 thành viên
- 0.5 điểm : SV có thể sử dụng bất kỳ cấu trúc dữ liệu nào đã học (mảng, danh sách liên kết, bảng băm, cây, đồ thị, ...) để cụ thể hoá ý tưởng của mình thông qua code hoặc mô tả chính xác cách hiện thực.

- Ví dụ một cách sử dụng cấu trúc mảng :

+ Cấu trúc lưu trữ thông tin người : 0.125 điểm

```
struct person
{
    int ID;
    char Name[];
}
```

+ Cấu trúc lưu trữ 1 post : 0.125 điểm

```
struct post
{
    char Content[]; // nội dung post
    char Date[]; // thời gian post
    int Count_like; // số like của post
    int ID_like[100]; // danh sách ID “bạn” đã like
}
```

+ Cấu trúc lưu trữ 1 thành viên : 0.125 điểm

```
struct member
{
    person PE; // thông tin thành viên
    post PO[1000]; // danh sách post
    int Number_post; // số lượng post
    person Friends[100]; // danh sách “bạn”
    int Number_friends; // số lượng “bạn”
}
```

+ Danh sách thành viên trong mạng : 0.125 điểm

```
member List[100];
```

- Ví dụ sử dụng cấu trúc danh sách liên kết :

- + SV có thể đề nghị sử dụng 100 danh sách liên kết ứng với 100 thành viên của mạng
- + Mỗi danh sách có ID riêng tương ứng với 1 thành viên: các phần tử trong danh sách là các “bạn” của ID.
- + Danh sách các post của mỗi thành viên có thể sử dụng tương tự cách sử dụng cấu trúc mảng

- Ví dụ sử dụng cấu trúc đồ thị :
 - + SV có thể đề nghị sử dụng 1 đồ thị, trong đó cạnh là thể hiện mối quan hệ giữa 2 thành viên trong mạng có là “bạn” hay không (giá trị 0 hoặc 1 - ma trận kề), đỉnh đồ thị lưu thông tin của 1 thành viên
 - + Danh sách các post của mỗi thành viên có thể sử dụng tương tự cách sử dụng cấu trúc mảng
 - Cách làm khác do SV đề nghị, Thầy, Cô thấy hợp lý
- b. Mô tả các giải thuật (các bước thực hiện giải thuật, không cài đặt) để hiện thực các chức năng của mạng đã mô tả ở phần trên và phù hợp với cấu trúc dữ liệu bạn đã đề nghị trong câu 4.a.

Thang điểm đề nghị :

- 0.25 điểm : SV qua cách trả lời của mình : nhận diện được / xác định được mục tiêu trong các chức năng của câu hỏi:
 - **"kết bạn"**: thực hiện cập nhật thêm danh sách “bạn” của 1 thành viên
 - **“xem thông tin”** : thực hiện duyệt qua danh sách các post mới nhất của “bạn” thành viên
 - **“like một post”** : thực hiện cập nhật trong post của “bạn” có thêm 1 thành viên đã like
- 0,5 điểm : SV mô tả cụ thể ý tưởng các bước cho các chức năng (hoặc qua code) :

+ Chức năng “kết bạn” : 0.125 điểm

Đầu vào : ID của thành viên, ID của “bạn” mới (cần kết bạn)

Đầu ra : Thông báo đã kết bạn thành công

Ý tưởng :

- Bước 1 : Kiểm tra ID “bạn” mới chưa có trong danh sách “bạn”
- Bước 2 : Kết bạn : cập nhật trong danh sách “bạn” của mình có thêm ID của “bạn” mới (thông qua cấu trúc mảng, danh sách liên kết, đồ thị,...SV đã sử dụng)
- Bước 3 : Thông báo đã kết bạn thành công.

+ Chức năng “like một post” : 0.25 điểm

Đầu vào : ID thành viên, Danh sách “bạn”

Đầu ra : Đã cập nhật like post của “bạn” thành công

Ý tưởng :

- Bước 1 : Hiện thị danh sách các post trong danh sách “bạn” của thành viên (thông qua cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 2 : Duyệt qua lần lượt các post cho đến hết danh sách post, nếu thích post nào đó chuyển qua bước 3, ngược lại qua bước 4
- Bước 3 : Like một post : cập nhật đã like, tăng số like cho post của “bạn” qua ID của “bạn” (thông qua truy xuất cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 4 : Quay lại bước 2

+ Chức năng “duyet các post” : 0.125 điểm

Đầu vào : ID thành viên, Danh sách “bạn”

Đầu ra : Đã duyệt các post của “bạn” thành công

Ý tưởng :

- Bước 1 : Hiển thị danh sách các post trong danh sách “bạn” của thành viên (thông qua cấu trúc mảng, danh sách liên kết, đồ thị,... SV đã sử dụng)
- Bước 2 : Duyệt qua lần lượt các post cho đến hết danh sách post

HẾT